Orcas Infrastructure

Bhavin Gandhi

Context

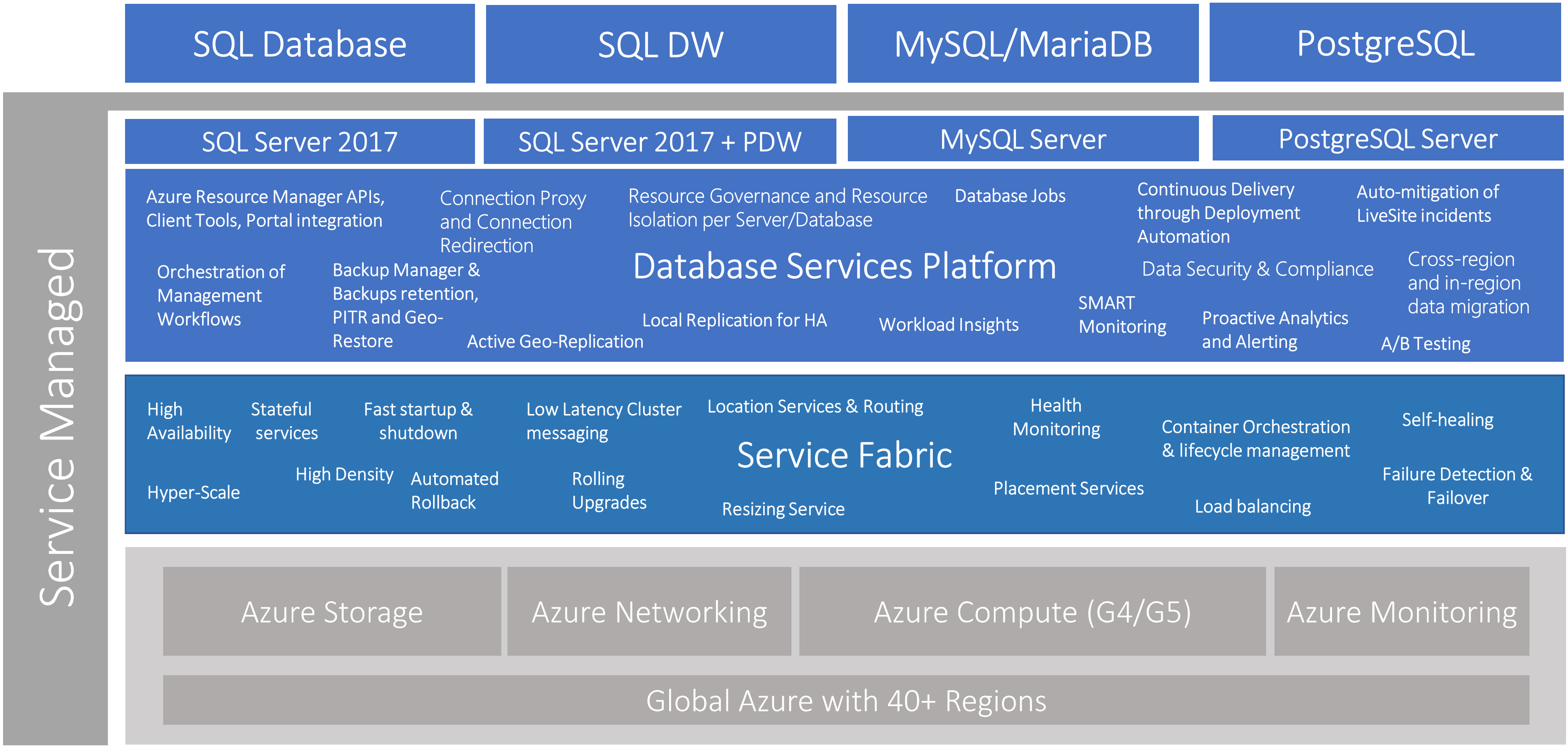
- Relational Database Service in Azure
  - Part of our developer story for bringing customers and their apps on our cloud platform

- Provision and Manage cloud databases for developers/DBAs/CSVs
  - High Availability (HA)
  - Automatic Backups/On-demand Point-In-Time Restores (PITR)
  - Disaster Recovery (DR),
  - Performance tuning/Recommendations
  - Security/Compliance/Auditing
  - On-demand scale up/down
  - Secondary read replicas, etc. etc.
  - **all while saving costs for customers and us**

Context

- Expand our Cloud Database offering to open source database engines
  - Offer choice to our customers for bringing their apps on Azure
- Started as an Incubation project with PostgreSQL as the first PoC and soon MySQL, and now MariaDB and possibly others.
- Leverage existing SQL DB investments (a.k.a Sterling platform) instead of starting from scratch
- Formal name for these services:
  - Azure Database for [PostgreSQL | MySQL | MariaDB]
  - https://azure.microsoft.com/en-us/status/

Azure Relational Database Platform



SQL Database

SQL DW

MySQL/MariaDB

PostgreSQL

Service Managed

SQL Server 2017

SQL Server 2017 + PDW

MySQL Server

PostgreSQL Server

Database Services Platform

Azure Resource Manager APIs, Client Tools, Portal integration

Connection Proxy and Connection Redirection

Resource Governance and Resource Isolation per Server/Database

Database Jobs

Continuous Delivery through Deployment Automation

Auto-mitigation of LiveSite incidents

Orchestration of Management Workflows

Backup Manager & Backups retention, PITR and Geo-Restore

Data Security & Compliance

Cross-region and in-region data migration

SMART Monitoring

Local Replication for HA

Workload Insights

Proactive Analytics and Alerting

Active Geo-Replication

A/B Testing

Service Fabric

High Availability

Stateful services

Fast startup & shutdown

Low Latency Cluster messaging

Location Services & Routing

Health Monitoring

Container Orchestration & lifecycle management

Self-healing

Hyper-Scale

High Density

Automated Rollback

Rolling Upgrades

Resizing Service

Placement Services

Load balancing

Failure Detection & Failover

Azure Storage

Azure Networking

Azure Compute (G4/G5)

Azure Monitoring

Global Azure with 40+ Regions

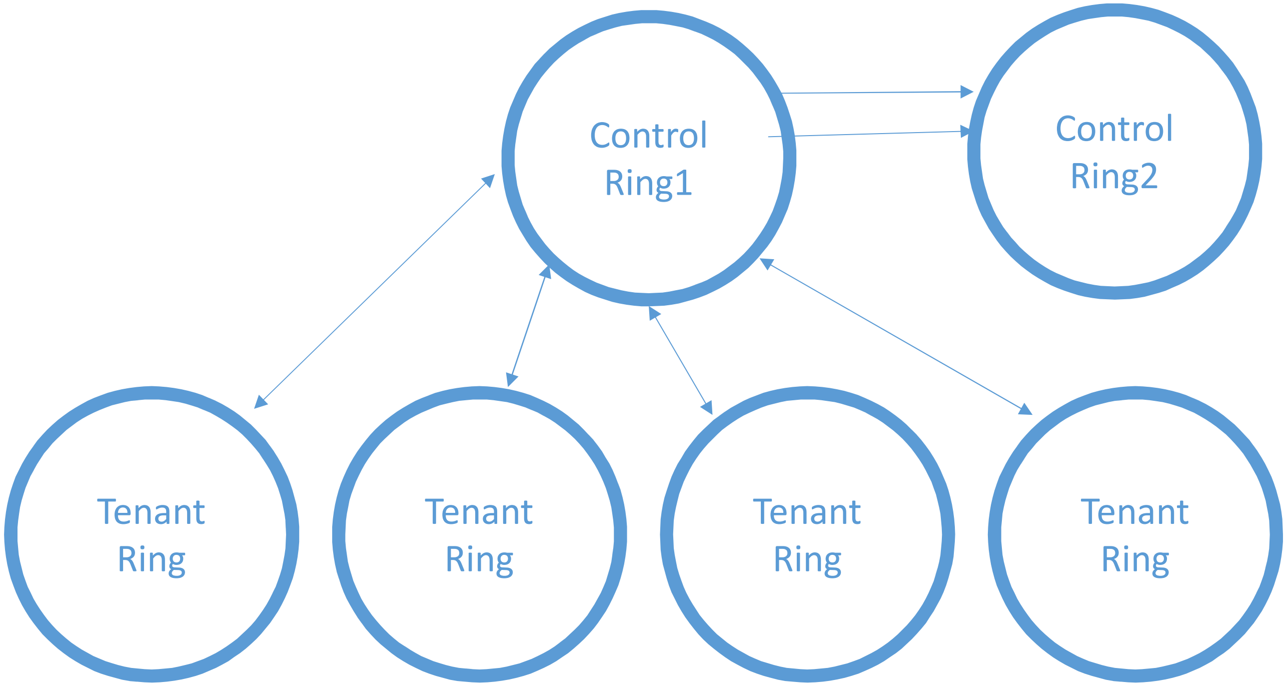
Customer Experience

- Manageability
  - CRUD operations on servers/databases
  - Firewall settings/VNET endpoints
  - Server logs
  - Threat detection
  - Performance Insights
  - Typically done through Azure Portal (portal.azure.com) or Azure CLI

- Connectivity
  - DB Admin tools/data applications connecting to databases to run queries/workload (OLTP)
  - psql/pgAdmin, workbench, or language specific drivers such as libpq, NpgSQL ADO.Net driver, ODBC connectors etc.

- **These two distinct experiences are implemented through different services in the architecture stack**

SQL DB Cluster (Region)



Cluster name:
wasd-prod-westeurope1-a

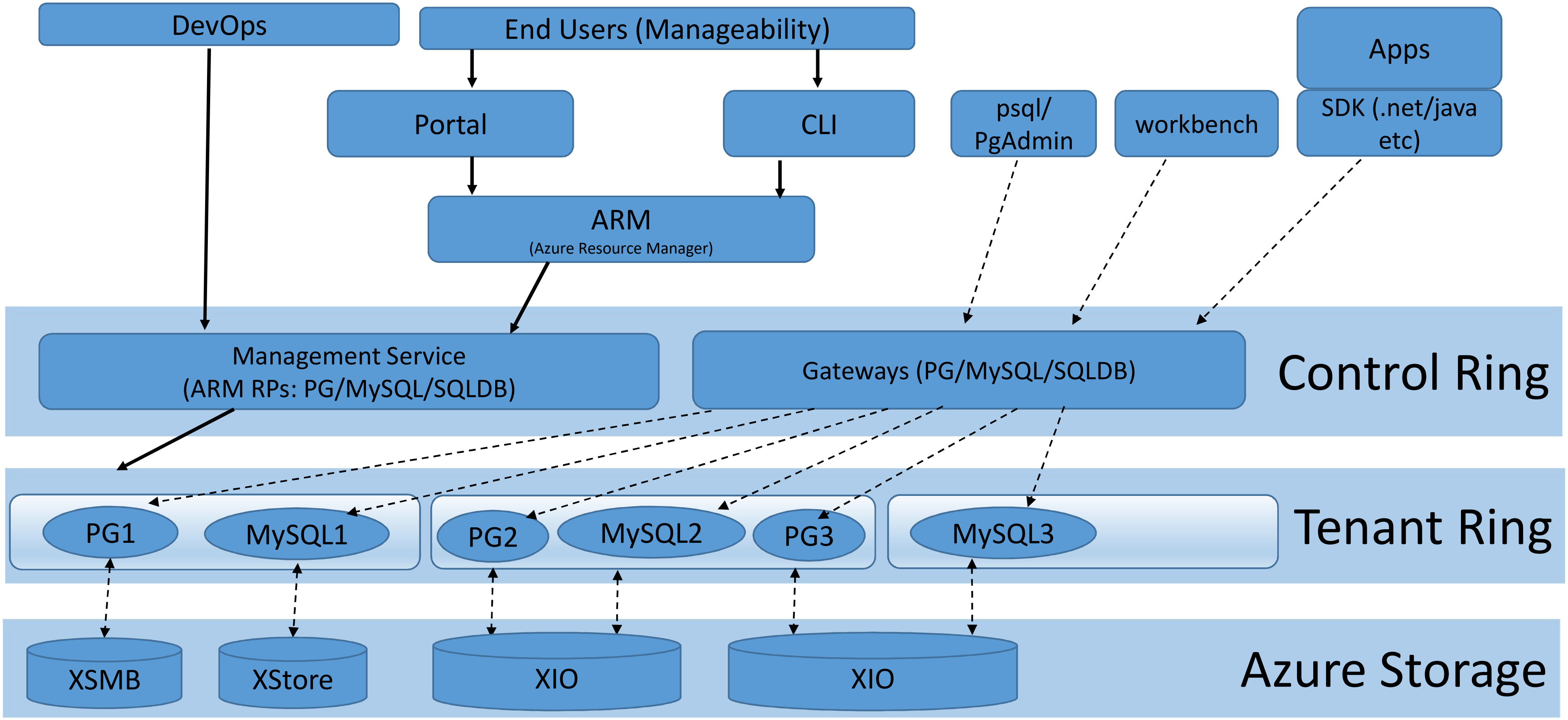
WASD – Windows Azure SQL Database

Prod – Environment (others test, stage)

Region<x> - xth cluster in that geography

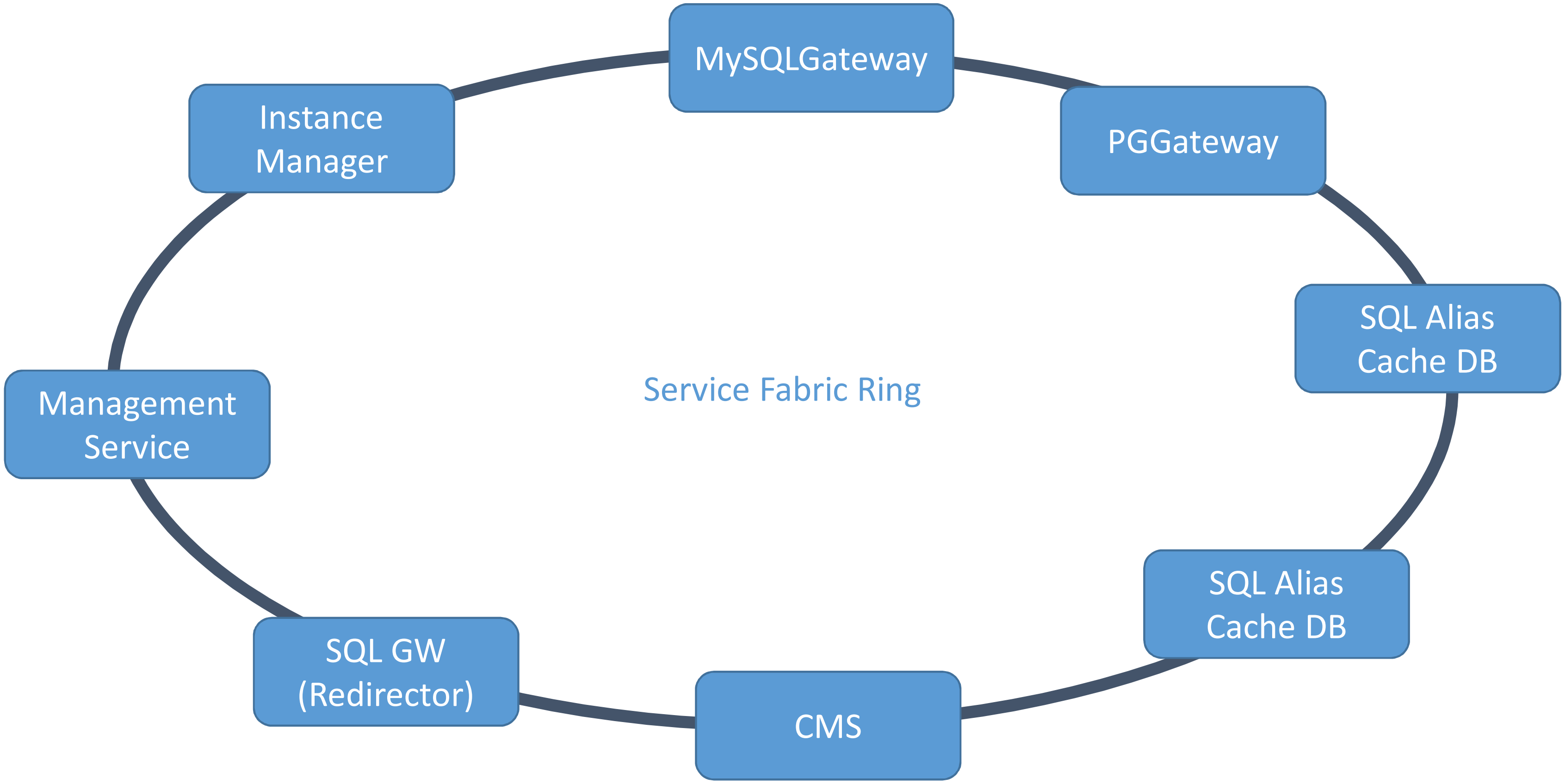
a/b – re-build of a cluster

High level Architecture



SFE View

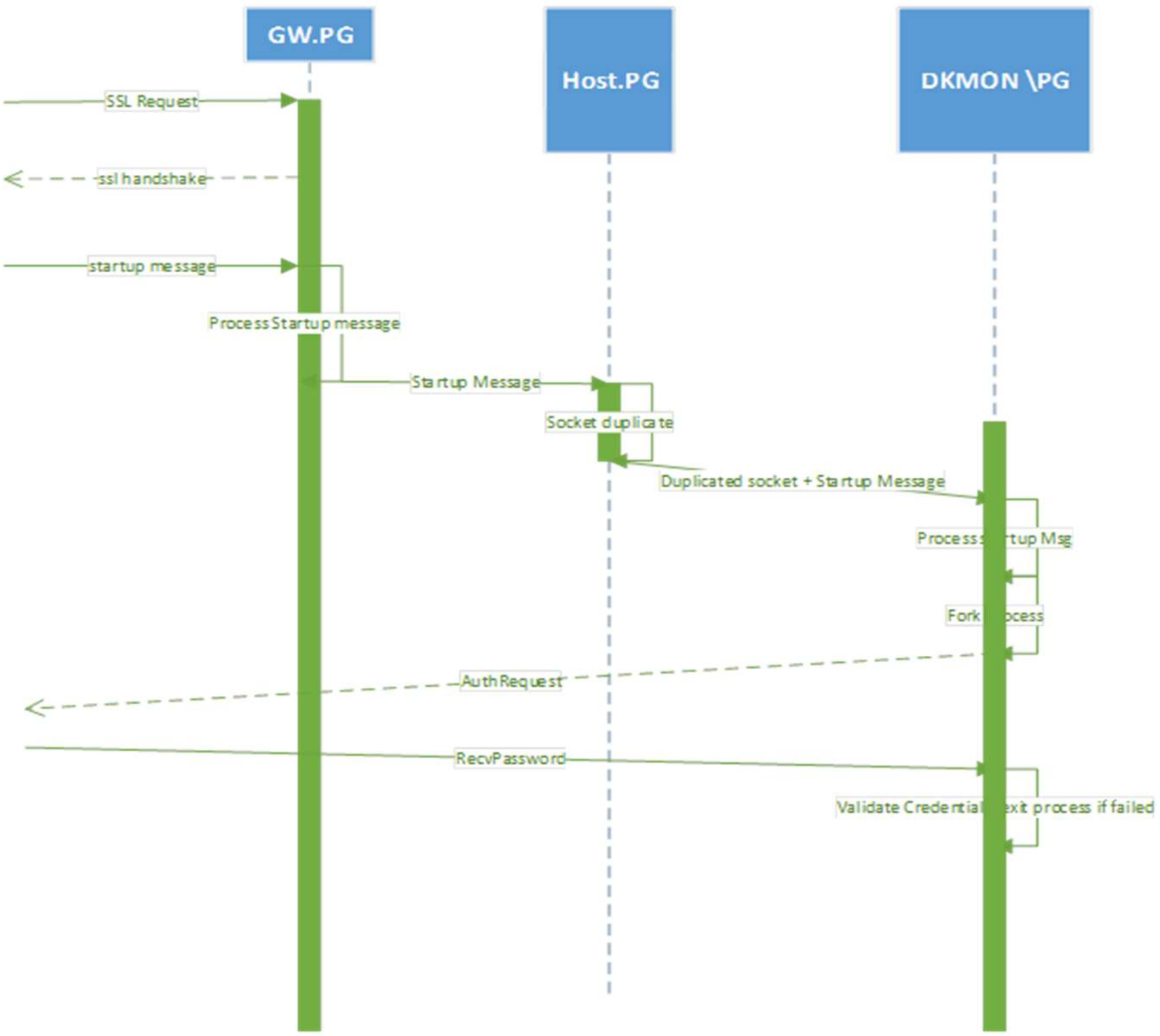
Control Ring Services



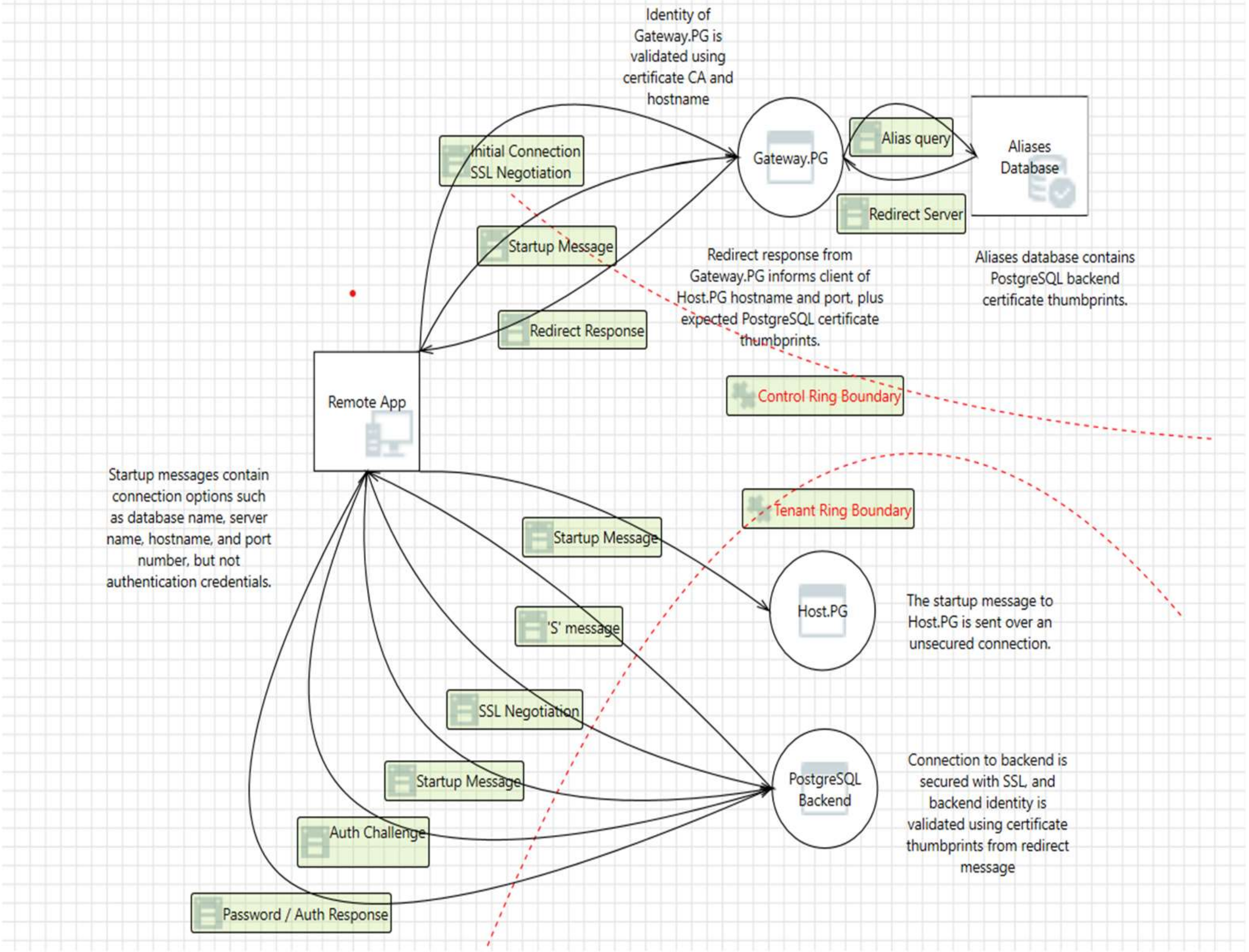
What does the Control Plane Do?

- Handles SQL/PG/MySQL Logins (Gateway services)
- Acts as the resource provider for ARM (ManagementService)
- Knows how to provision SQL Logical servers, PG/MySQL Servers
- CRUD operations
  - **C**reate: Create a PG server
  - **R**ead: Get the list of servers/databases
  - **U**pdate: Change the size (SLO/Storage) of this server
  - **D**elete: Delete this server
- Also orchestration of lots and lots of other workflows (PITR, Deployment, Security, Workload Insights, Geo-Restore, Deactivation, etc.)

Login Flow (Default)



Login Flow (Redirection)



Identity of Gateway.PG is validated using certificate CA and hostname

Redirect response from Gateway.PG informs client of Host.PG hostname and port, plus expected PostgreSQL certificate thumbprints.

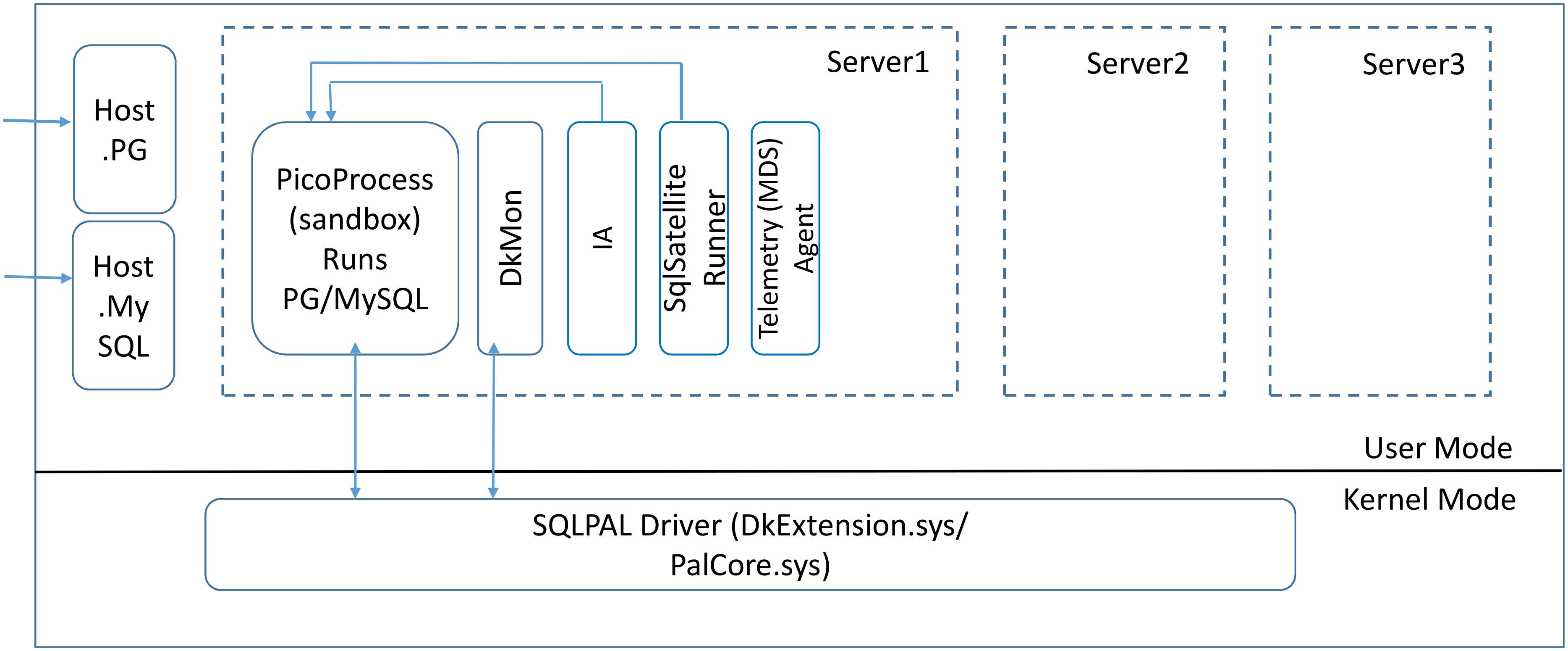
Aliases database contains PostgreSQL backend certificate thumbprints.

Startup messages contain connection options such as database name, server name, hostname, and port number, but not authentication credentials.

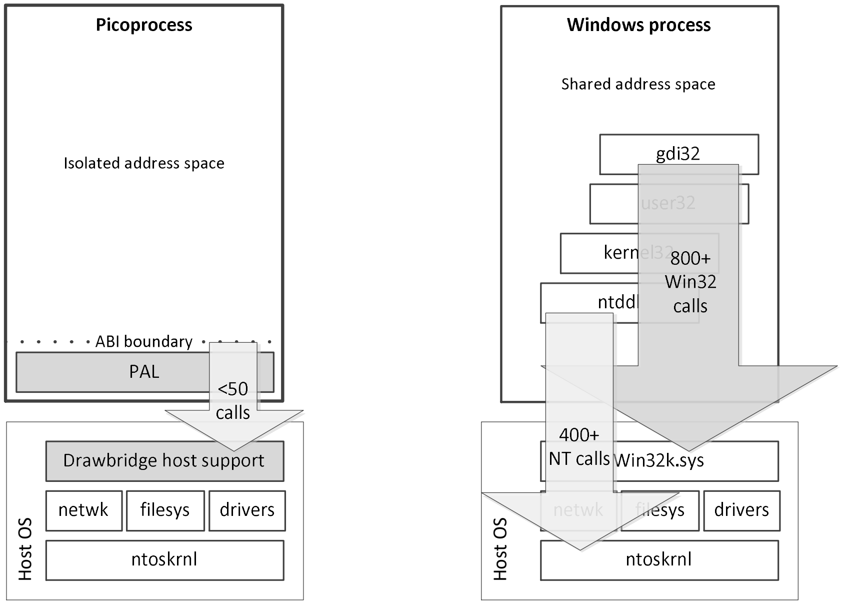
The startup message to Host.PG is sent over an unsecured connection.

Connection to backend is secured with SSL, and backend identity is validated using certificate thumbprints from redirect message

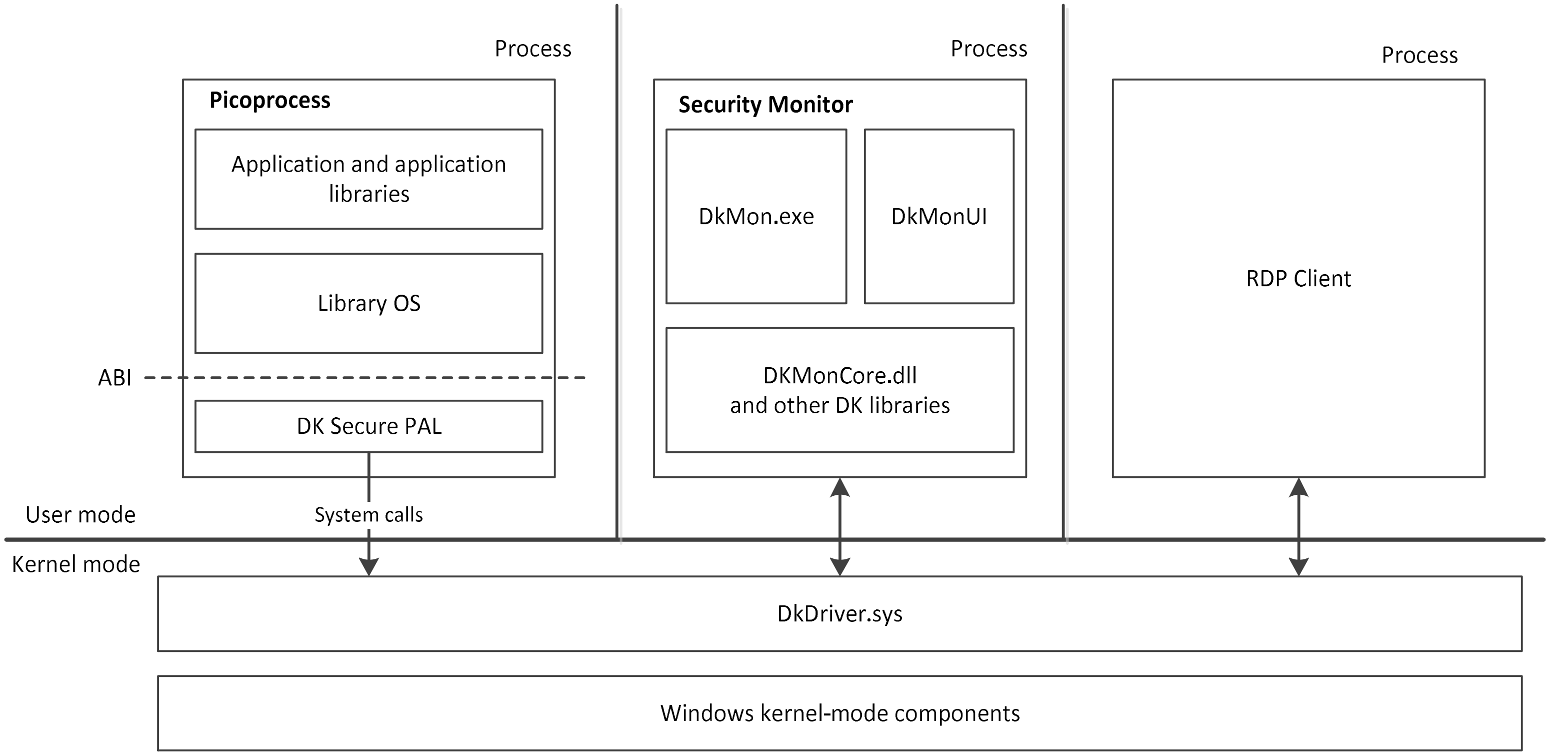
Tenant Ring (DB node view)



SQLPAL

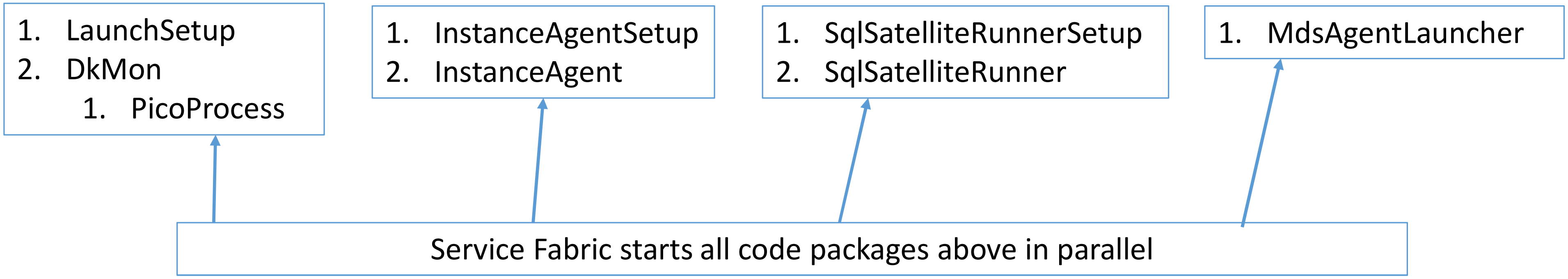

Picoprocess
Isolated address space
ABI boundary
PAL
<50 calls
Drawbridge host support
netwk
filesys
drivers
ntoskrnl
Host OS
Windows process
Shared address space
gdi32
user32
kernel32
ntdll
800+ Win32 calls
400+ NT calls
Win32k.sys
netwk
filesys
drivers
ntoskrnl
Host OS

SQLPAL


Process
Picoprocess
Application and application libraries
Library OS
ABI
DK Secure PAL
Process
Security Monitor
DkMon.exe
DkMonUI
DKMonCore.dll and other DK libraries
Process
RDP Client
User mode
System calls
Kernel mode
DkDriver.sys
Windows kernel-mode components

Server startup

- Service fabric orchestrates starting up all services (and restarting them in the event of planned/unplanned failures)
- Below 4 code packages



1. LaunchSetup
2. DkMon
    1. PicoProcess

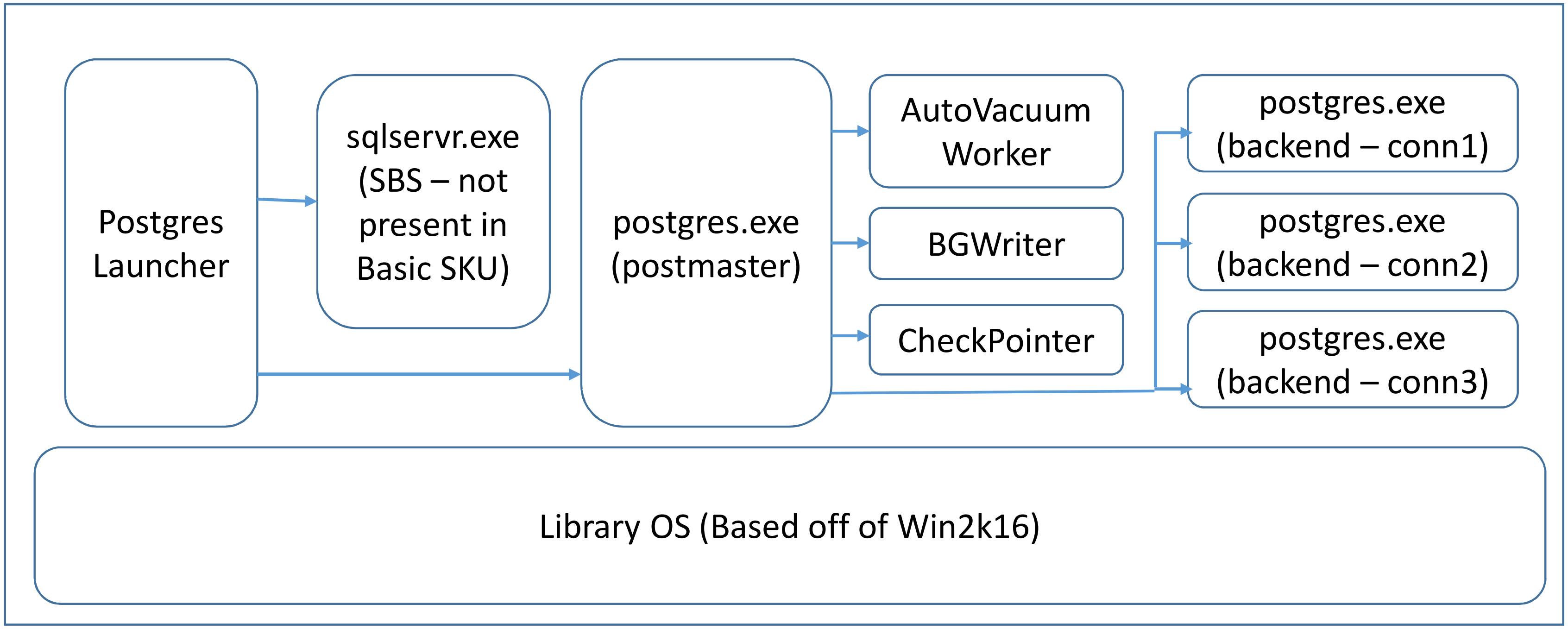
1. InstanceAgentSetup
2. InstanceAgent

1. SqlSatelliteRunnerSetup
2. SqlSatelliteRunner

1. MdsAgentLauncher

Service Fabric starts all code packages above in parallel

PicoProcess View (for PostgreSQL)



Postgres Launcher

sqlservr.exe (SBS – not present in Basic SKU)

postgres.exe (postmaster)

AutoVacuum Worker

BGWriter

CheckPointer

postgres.exe (backend – conn1)

postgres.exe (backend – conn2)

postgres.exe (backend – conn3)

Library OS (Based off of Win2k16)

Useful Telemetry Tables (Kusto)

- MonManagement (For Manageability requests)
- MonLogin (filter by AppTypeName => Gateway.PG/Gateway.MySQL/Host.PG/Host.MySQL)
- MonRdmsPgSqlLaunchSetup (LaunchSetup for PG)
- MonRdmsPgSqlSandbox (DkMon & PG Picoprocess telemetry)
- MonRdmsMySqlLaunchSetup (LaunchSetup for MySQL)
- MonRdmsMySqlSandbox (DkMon & MySQL Picorprocess telemetry)
- MonRdmsInstanceAgent (for IA logs)